

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

## Zadání bakalářské práce

Student:

**Jiří Seidl**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: GIRITON Systems s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Marek Běhálek, Ph.D.**

Konzultant bakalářské práce: Ing. Jan Gřeš

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

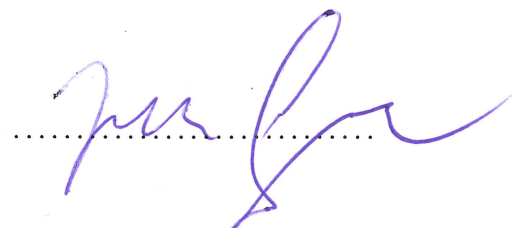
Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016

.....  
Girdl

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 29. dubna 2016



Rád bych na tomto místě poděkoval zejména svému konzultantovi Ing. Janu Gřešovi, MSc. za za umožnění celé bakalářské praxe ve firmě GIRITON Systems s.r.o a za jeho trpělivost, kterou se mnou měl. A dále bych chtěl poděkovat svému vedoucímu bakalářské práce Ing. Markovi Běhálkovi, Ph.D. za rady a připomínky při jejím vypracování.

## **Abstrakt**

Tato bakalářská práce pojednává o mé odborné praxi ve firmě GIRITON Systems s.r.o. Nejprve představím firmu samotnou, a také její produkty. Potom popíšu, jaká byla má náplň práce a kolik jsem tomu věnoval času. Nakonec shrnu své dojmy z praxe, co jsem se během ní naučil a jaké vědomosti ze školy jsem při tom uplatnil nebo co mi naopak zcházelo.

**Klíčová slova:** bakalářská práce, GIRITON Systems s.r.o., Java, Android, webová aplikace

## **Abstract**

This bachelor thesis describes my professional practice in a company GIRITON Systems s.r.o. I will introduce the company and its products first. Then i describe my workload and how much time it took me. Last I summarize my impressions from the practice, what I have learned in the course and what knowledge from a college i used or what i missed on the other hand.

**Key Words:** bachelor thesis, GIRITON Systems s.r.o., Java, Android, web application

# Obsah

Seznam použitých zkratek a symbolů	8
Seznam obrázků	9
Seznam tabulek	10
<b>1 Úvod</b>	<b>11</b>
<b>2 O firmě</b>	<b>12</b>
2.1 Docházka GIRITON . . . . .	12
<b>3 Seznam úkolů</b>	<b>16</b>
3.1 Android Píchací Hodiny . . . . .	16
3.2 Webové Píchací Hodiny . . . . .	18
3.3 Moje Docházka . . . . .	20
3.4 Další úkoly . . . . .	22
<b>4 Závěr</b>	<b>23</b>
4.1 Získané znalosti . . . . .	23
4.2 Zhodnocení praxe . . . . .	23
<b>Literatura</b>	<b>25</b>

## Seznam použitých zkratek a symbolů

CSS	– Cascading Style Sheets
DOM	– Document Object Model
GPS	– Global Positioning System
GUI	– Graphical User Interface
HTML	– Hyper Text Markup Language
NFC	– Near Field Communication
QR code	– Quick Response Code
REST API	– Representational State Transfer Application Programming Interface
RFID	– Radio Frequency Identification
XML	– Extensible Markup Language



## Seznam obrázků

1	Logo firmy . . . . .	12
2	Diagram komponent systému Docházka GIRITON . . . . .	12
3	Webová Aplikace . . . . .	14
4	Aplikace Moje Docházka . . . . .	14
5	Aplikace Píchací Hodiny pro Android . . . . .	15
6	Aplikace Píchací Hodiny pro Web . . . . .	15

## Seznam tabulek

1	Odhad času stráveného na jednotlivých úkolech v aplikaci Android Píchací Hodiny	18
2	Odhad času stráveného na jednotlivých úkolech v aplikaci Webové Píchací Hodiny	19
3	Odhad času stráveného na jednotlivých úkolech v aplikaci Moje Docházka . . . .	21
4	Odhad času stráveného na dalších úkolech . . . . .	22

# 1 Úvod

K vypracování bakalářské práce formou odborné praxe ve firmě mě vedla touha získat zkušenosti s vývojem reálného projektu v týmu lidí už při studiu, které doufám zúročím později na pracovním trhu. Praxi jsem vykonával ve firmě GIRITON Systems s.r.o.[1] Pro tuto firmu jsem se rozhodl, hlavně protože nabízeli pozici Java programátora. V té době jsem byl po předchozích zkušenostech s C a C++ z Javy velmi nadšený a napsal jsem v ní 5 jednoduchých her. Ve zdokonalování svých dovedností v tomto jazyce jsem chtěl pokračovat, a proto pro mě byla nabízená praxe atraktivní. Dále pro mě bylo podstatné geografické umístění firmy přímo v areálu školy, zhruba 5 minut chůze od studentských kolejí, kde jsem byl ubytován.

V následujících kapitolách představím firmu a její produkty a následně popíšu svoji práci na jednotlivých produktech a částech celého systému. Zmíním problémy, na které jsem při tom narazil, jak jsem je řešil, proč jsem zvolil dané řešení a pokusím se také odhadnout časovou náročnost své práce. Na závěr pak shrnu, co jsem se při tom naučil, v čem si myslím, že pro mě byla praxe přínosem a jaké vědomosti ze školy jsem při řešení daných problémů uplatnil nebo co mi naopak chybělo.

## 2 O firmě

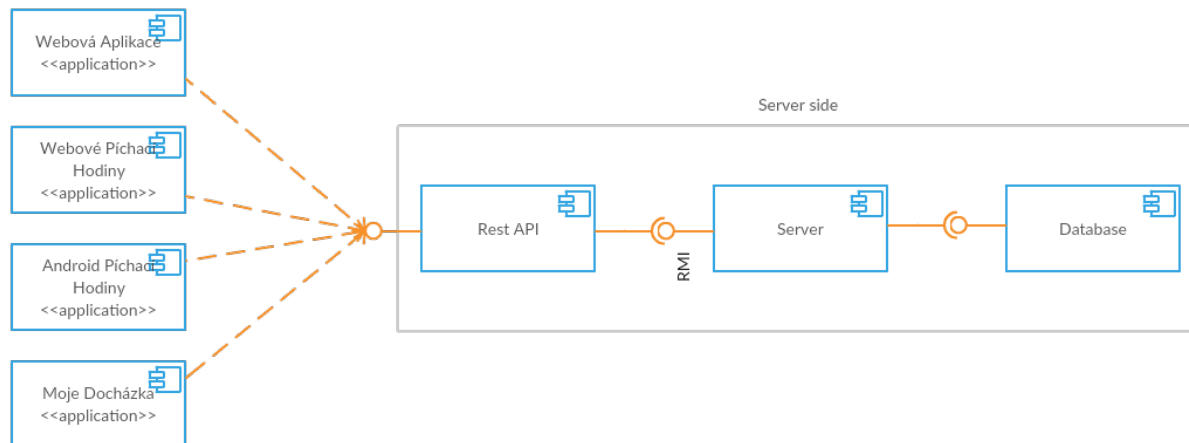
Firma GIRITON Systems s.r.o. je mladá firma zabývající se tvorbou podnikových informačních systémů se sídlem v Podnikatelském inkubátoru Vysoké školy báňské - Technické univerzity v Ostravě. Soustředí se na moderní cloudové systémy a jejich propojení s mobilními telefony a tablety. Vlajkovou lodí firmy je informační systém Docházka GIRITON.



Obrázek 1: Logo firmy

### 2.1 Docházka GIRITON

Jde o moderní Cloudový informační systém na sledování docházky zaměstnanců. Systém, kromě sledování docházky, obsahuje spoustu dalších funkcí pro správu zaměstnanců a řízení výroby. Sestává ze serverové části a několika aplikací, které zde krátce představím. Jejich vzájemné propojení je vidět na obrázku 2.



Obrázek 2: Diagram komponent systému Docházka GIRITON

### 2.1.1 Webová Aplikace

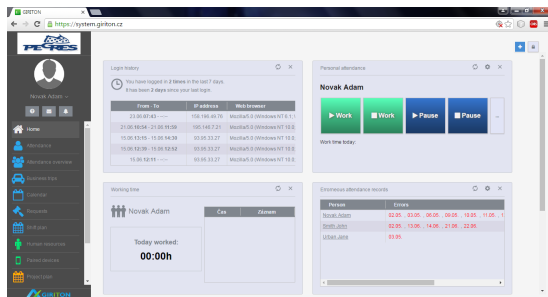
Představuje hlavní terminál pro správu celého systému. Zvládá všechny myslitelné operace. Podle oprávnění uživatele, který se do ní přihlásí, umožňuje od běžného vkládání a prohlížení vlastní docházky, až po nastavování celého systému, vytváření nových aktivit nebo nastavování oprávnění ostatním uživatelům. Celý systém je velice flexibilní a zkrz tuto aplikaci, si jej uživatel může nastavit přesně podle svých specifických požadavků firmě na míru, nebo dokonce vylepšit organizaci firmy, s využitím některého z mnoha nástrojů, které systém nabízí.

### 2.1.2 Moje Docházka

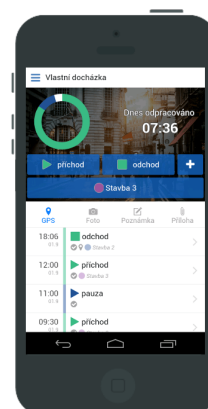
Aplikace je určena pro vkládání docházky přímo z mobilních telefonů zaměstnanců, což je užitečné například pro služební cesty, kdy může zaznamenávat polohu zaměstnance pomocí GPS. K záznamům docházky je také možnost připojit fotky, poznámky nebo soubory. Dokáže pracovat i v offline režimu a ukládat záznamy do telefonu aby je pak po připojení k internetu mohla odeslat na server. Je napsána v JavaScriptu, HTML a CSS s využitím frameworku AngularJS[2] a Onsen UI[3]. Pomocí Apache Cordova[4] se pak exportuje pro platformy Android, iOS a Windows Phone.

Aplikace dále umožňuje:

1. Čipování příchodů a odchodů práce, pauzy, dovolené, nemocenské a dalších aktivit které si zaměstnavatel může nadefinovat ve Webové Aplikaci.
2. Zaznamenávat práci na jednotlivých projektech.
3. prohlížet historii vlastní docházky s přehledným grafickým znázorněním.
4. prohlížet docházku jiných osob pokud k tomu mám oprávnění.
5. prohlížet které osoby jsou aktuálně na pracovišti.
6. vytvářet, prohlížet a pokud k tomu mám oprávnění, tak i schvalovat žádosti.
7. prohlížet plán směn svůj a pokud jsem k tomu oprávněn, tak i ostatních osob.
8. vytvářet a prohlížet záznamy o služební cestě s informacemi o pozici, služebním vozidle, stavu tachometru a paliva.



Obrázek 3: Webová Aplikace



Obrázek 4: Aplikace Moje Docházka

### 2.1.3 Píchnací Hodiny

Tato aplikace má 2 varianty:

- Nativní aplikace pro systém Android psaná v Javě.
- Webová aplikace psaná v JavaScriptu, HTML a CSS.

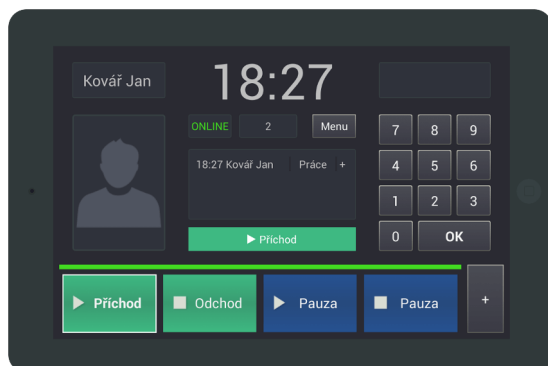
Obě verze jsou téměř totožné. Aplikace jsou určeny k tomu, aby běžely na tabletu umístěném někde u vchodu do zaměstnání nebo na dotykové pokladně, a aby umožňovaly snadné čipování docházky zaměstnancům. To lze provést hned několika způsoby.

1. Pomocí pinu zadaného z klávesnice.
2. Pomocí QR kódu sejmutého kamerou.
3. Pomocí magnetických karet.
4. Pomocí bezkontaktního RFID nebo NFC čipu.

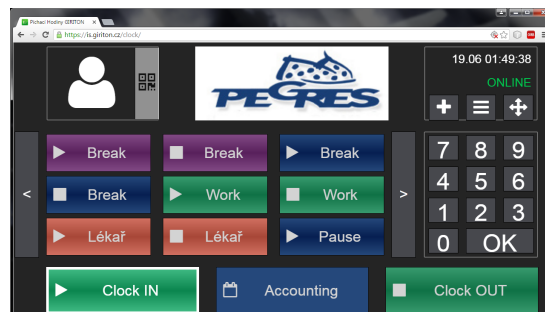
Při zaznamenání jednotlivých událostí je volitelně také možno pořídit fotografii dané osoby a zaznamenat GPS pozici. Aplikace dále umožňuje:

1. Čipování příchodů a odchodů práce, pauzy, dovolené, nemocenské a dalších aktivit které si zaměstnavatel může nadefinovat ve Webové Aplikaci.
2. Zaznamenávat práci na jednotlivých projektech.
3. prohlížet historii vlastní docházky s přehledným grafickým znázorněním
4. prohlížet docházku jiných osob pokud k tomu mám oprávnění
5. prohlížet které osoby jsou aktuálně na pracovišti

6. vytvářet, prohlížet a pokud k tomu mám oprávnění, tak i schvalovat žádosti
7. prohlížet plán směn svůj a pokud jsem k tomu oprávněn, tak i ostatních osob



Obrázek 5: Aplikace Píchnací Hodiny pro Android



Obrázek 6: Aplikace Píchnací Hodiny pro Web

#### 2.1.4 Pracovní zařazení studenta

Do firmy jsem nastoupil na pozici Java programátora. Rozšiřoval a upravoval jsem Android verzi Píchnacích hodin a opravoval případné chyby. Později jsem se přesunul na webovou verzi Píchnacích hodin a aplikaci Moje Docházka u kterých jsem zůstal až do konce praxe. Podrobněji svou práci na jednotlivých aplikacích rozepíšu v následující kapitole.

## 3 Seznam úkolů

Ze zadaných úkolů jsem vybral ty časově náročnější a u těch krátce představuji problém a jak jsem ho řešil.

### 3.1 Android Píchací Hodiny

V této aplikaci jsem měl za úkol:

- Vytvořit novou agendu pro podávání a schvalování žádostí zaměstnanců.

Tento úkol mi zabral asi nejvíce času, a to hlavně díky velkému množství podúkolů, které v sobě zahrnuje. Musel jsem vytvořit novou obrazovku s navigační lištou a třemi panely, které lze přepínat. Pro každý panel bylo potřeba vytvořit samostatné rozložení prvků a naplnit je daty, která jsem získal voláním požadavků na Rest API serveru. Na první panel jsem umístil seznam nových žádostí a tzv. „Floating Action Button”[5], který vyvolává dialog s formulářem pro přidání nové žádosti. Druhý panel obsahuje seznam schválených a zamítnutých žádostí s indikací jejich stavu a na třetím panelu se zobrazuje seznam žádostí, které jako uživatel mohu schválit. Každá žádost na schválení otevírá další dialog s informacemi o dané žádosti a akcemi které s ní lze provádět.

- Rozšířit stávající agendu pro prohlížení docházky tak, aby zobrazovala denní a měsíční sumy docházky.

Na tomto úkolu jsem spolupracoval s kolegou. Cílem bylo aby v seznamu dnů s grafy docházky byly dostupné i celkové sumy časů strávených jednotlivými aktivitami a aby v agendě byly dostupné sumy za celý měsíc. Vzhledem k proměnlivosti počtu aktivit pro které se sumy mohou zobrazovat, jsme každému dni přidali druhý řádek, který se zobrazí jen u aktuálně vybraného dne a bude obsahovat všechny potřebné informace. Pro měsíční sumy, jsme pak zvolili výsuvný panel na spodním okraji obrazovky.

- Vytvořit vlastní komponentu tlačítka, která by podporovala obrázkové ikony spolu s textovým popiskem.

V aplikaci jsme na mnoha místech používali tlačítka s textem a přes něj relativně pozicovanou Font Awesome[6] ikonku, protože taková komponenta v dostupných knihovnách chyběla. To bylo nepraktické, a proto bylo třeba vytvořit komponentu speciálně pro naše požadavky s možností nastavení různých parametrů. A měla jít použít v XML souboru s rozvržením GUI, bez nutnosti nastavovat font v java kódu při každém použití.

- Přidat na všechny tlačítka a klikatelné plochy tzv. „ripple effect”[7] pro zvýšení responzivity GUI.



Zde bylo největším problémem zachování podpory starších verzí systému Android. Po prozkoumání možností jsem se rozhodl, použít k tomu knihovnu „Android L Ripple effect wrapper for Views”[8] od GitHub uživatele Balysv.

- Deaktivovat některé volby v nastavení aplikace v závislosti na stavu jiných voleb.

Uživatel si může zvolit ze 2 variant rozložení hlavní obrazovky. Pokud si vybere jedno, tak je zbytečné, mu umožňovat volby, které se projeví jen na druhém rozložení. Taktéž je zbytečné mu umožňovat například nastavení rozlišení kamery v případě, že žádnou nemá. Je ale žádoucí, aby uživatel viděl, že aplikace podporuje tyto funkcionality. Proto jsem zvolil pouze deaktivování těchto voleb, místo jednoduchého skrytí.

- Zobrazit softwarovou klávesnici, i když je připojená hardwarová klávesnice.

Nově vytvořená agenda Žádosti umožňovala přidat komentář. V systému Android se však čtečka čipů jeví jako externí klávesnice a díky tomu se nezobrazuje tlačítko, pro zobrazení softwarové klávesnice na obrazovce. Cílem tedy bylo, umožnit zobrazení této klávesnice i v případě, že je připojená čtečka čipů. Tento úkol se mi bohužel nepodařilo vyřešit. Po dlouhém hledání a testování různých řešení jsem dospěl k závěru, že díky internímu bezpečnostnímu opatření systému Android nelze změnit algoritmus, který rozhoduje, zda se zobrazí nebo nezobrazí softwarová klávesnice. Řešení by tedy bylo možné jen u tzv. „rootnutých”[9] zařízení, a proto jsme od této funkcionality museli upustit.

- Vytvořit automatické testy pro nové nebo aktualizované agendy a vytvářet při tom screenshoty obrazovky.

Cílem bylo automatizovat testování aplikace, aby bylo možno testovat aplikaci na několika zařízeních najednou, bez nutnosti zásahu člověka. Dále bylo žádoucí při tom pořizovat snímky obrazovky v jednotlivých agendách, které by pak bylo možno nahrát na web a do Google Play[10]. Problém jsem měl pouze s mazáním nastavení, které si aplikace do telefonu ukládala, před zahájením samotného testování. Změny se totiž projeví vždy až při příštím spuštění aplikace. To jsem nakonec vyřešil přidáním dalšího testu jakoby na nultou pozici v pořadí vykonávaných testů, který měl za úkol pouze vymazat data aplikace a restartovat ji.

- A dále značné množství drobnějších úprav GUI i chování aplikace a oprav chyb.

Tabulka 1: Odhad času stráveného na jednotlivých úkolech v aplikaci Android Píchací Hodiny

Úkol	Strávený čas [h]
Vytvořit agendu Žádosti	80
Rozšířit Přehled docházky o sumy	20
Vytvořit komponentu tlačítko	20
Přidat „ripple effect”	20
Deaktivovat nastavení vypnutých funkcí	5
Zobrazit softwarovou klávesnici s čtečkou	30
Vytvořit automatické testy aplikace	20

### 3.2 Webové Píchací Hodiny

Zde byly mé úkoly následující:

- Po vzoru Android verze aplikace vytvořit novou agendu pro podávání a schvalování žádostí zaměstnanců.

Obdobně jako v Android verzi jsem musel vytvořit celé GUI, vytvořit metody pro komunikaci s Rest API serveru a získaná data zobrazit v jednotlivých záložkách. Nejproblematictější pro mě bylo asi pochopení navigátoru mezi obrazovkami a nastavení tak, aby při kliku na tlačítko zpět přeskakoval některé obrazovky podle toho, odkud byly vyvolány. Místo dialogů se totiž v této verzi vkládají nové snímky s odpovídajícími přechody. Bylo žádoucí, aby se při kliku na tlačítko pro vytvoření nové žádosti první zobrazila stránka s výběrem typu žádosti, a následně, po výběru některé z možností, se zobrazil požadovaný formulář. Z něj pak bylo možno znova vyvolat obrazovku s výběrem typu žádosti, ale tentokrát se při stisku tlačítka zpět mělo vrátit na formulář místo seznamu žádostí. Při stisku tlačítka zpět ve formuláři nebo při odeslání vyplněné žádosti se měla obrazovka s výběrem typu žádosti přeskočit. Řešení tohoto úkolu mi zabralo asi poloviční čas, ve srovnání se stejným úkolem v Android verzi aplikace, a to i přesto, že jsem se s vývojem webových aplikací teprve seznamoval. To bylo způsobeno především tím, že jsem danou agendu již jednou vytvářel, a také díky rychlejšímu vývoji webových aplikací už ze své podstaty, jelikož odpadá dlouhé čekání na zkompileování aplikace a její nahrání a spuštění na zařízení pokaždé, když chceme vidět, jak se projeví provedené změny.

- Úprava tzv. „carouselu”[11] tak, aby fungoval spolu s Electron frameworkem[12].

Carousel je obecně označována webová komponenta, která umožňuje plynule vyměňovat obsah části stránky pouze přetažením myši nebo automaticky v pravidelných intervalech. Doposud používaná komponenta z oblíbeného frameworku Onsen UI ovšem nefungovala správně, pokud jsme z webové aplikace chtěli vytvořit nativní desktopovou aplikaci pomocí frameworku Electron. Musel jsem tedy najít vhodnou náhradu, kterou byla knihovna Slick[13].

- Přidat do agendy Plán Směn možnost barevného šrafování a indikaci dostupnosti a poznámky u jednotlivých dnů.

V Plánu směn se vybarvují jednotlivé dny podle typu naplánované směny barvou, která mi chodí ze serveru. Bylo však nutné graficky rozlišit dny, na které má uživatel naplánováno více směn (například ranní a odpolední). To mělo zajistit šrafování ve kterém se střídaly barvy, svázané s konkrétními typy směn. Dále jsem měl zobrazovat v buňce kalendáře ikonku, pokud u naplánované směny byla nějaká poznámka a zelený nebo červený puntík v případě, že zaměstnanec oznamuje, kdy je nebo není dostupný a směny mu pak podle toho volí manažer.

- Deaktivovat některé volby v nastavení aplikace v závislosti na stavu jiných voleb.

Uživatel si může zvolit ze 2 variant rozložení hlavní obrazovky. Pokud si vybere jedno, tak je zbytečné, mu umožňovat volby, které se projeví jen na druhém rozložení. Taktéž je zbytečné mu umožňovat například nastavení rozlišení kamery v případě, že žádnou nemá. Je ale žádoucí, aby uživatel viděl, že aplikace podporuje tyto funkcionality. Proto jsem zvolil pouze deaktivování těchto voleb, místo jednoduchého skrytí.

- A v neposlední řadě značné množství drobnějších úprav GUI i chování aplikace a oprav chyb.

Tabulka 2: Odhad času stráveného na jednotlivých úkolech v aplikaci Webové Píhací Hodiny

Úkol	Strávený čas [h]
Vytvořit agendu žádosti	40
Upravit „carousel” pro framework Electron	20
Rozšířit Plán směn	5
Deaktivovat nastavení vypnutých funkcí	5

### 3.3 Moje Docházka

V této aplikaci bylo mým úkolem:

- Aktualizovat aplikaci pro verzi 2.0 frameworku Onsen UI.

Vyšla nová verze frameworku Onsen UI, která přinášela spoustu nových komponent a vlastností. Pro použití v naší aplikaci ovšem bylo nutné aktualizovat kód, protože některé komponenty se v nové verzi používaly trochu jinak. Po opravení většiny chyb, způsobených přechodem na novou verzi frameworku, mi zbyl jeden problém, který se mi nepodařilo vyřešit a časem jsem dospěl k závěru, že jde o zatím nenahlášenou chybu přímo v nové verzi Onsen UI. Vrátil jsem tedy provedené změny v aplikaci do původního stavu, a chybu jsem s detailním popisem a online ukázkou nahlásil autorům frameworku. Ti chybu opravili a vydali v další podverzi. Nicméně už jsem neměl čas dokončit celou aktualizaci, jelikož tou dobou se má praxe chýlila k závěru.

- Opravit chybu při zobrazování zaznamenané trasy služební cesty těsně po jejím ukončení.

Problém spočíval v tom, že po ukončení služební cesty, když jsem si zobrazil mapu se zaznamenanou trasou, tak jsem viděl jenom koncový bod. To bylo způsobeno rozdílnými unikátními identifikátory, pod kterými se 2 související záznamy (začátek a konec) ukládaly do cache úložiště. To se pak synchronizovalo ze serverem který si ze začátku a konce vytvořil jednu služebku, což řešilo problém s vkládáním párových záznamů z různých zařízení, a vrátil mi svoji verzi která přepsala moje dočasně uložené lokální záznamy.

- Vytvořit podporu na straně aplikace pro nové typy aktivit, počítadlo, poznámka a cena.

Na žádost klienta bylo potřeba naimplementovat typy aktivit, které se nepočítaly na časové úseky ale například jako celé číslo nebo poznámka s možností úpravy podle libosti. Vytvořil jsem tedy GUI i metody pro komunikaci s již připravenými Rest API metodami na serveru a to vedlo k dalšímu úkolu.

- Stahovat sumy časů jednotlivých aktivit a projektů pomocí nové Rest API metody.

Vzhledem k předchozímu úkolu již nedostačovaly metody pro získávání záznamů ze serveru, které obsahovaly čas začátku a konce aktivit, ze kterých se počítalo trvání. Bylo proto vytvořena nová, univerzálnější Rest API metoda pro získávání sum ze serveru a mým úkolem bylo přepracovat stahování a výpočet sum aktivit a projektů tak, aby využívala tuto novou metodu.

- Vytvořit novou obrazovku s historií záznamů.

Předchozí umístění tohoto seznamu zaujal seznam aktivit a proto bylo nutné ho přesunout do nové obrazovky přístupné přes tlačítko a rekonstruovat již smazané metody spjaté s touto funkcionalitou.

- Přepracovat tutoriál.

Tutoriál byl vytvořen jako série vzájemně provázaných stránek které každá inicializovala script pro tutoriál, registraci a přihlášení uživatele. Bylo tedy vhodné tutoriál přepracovat nějak inteligentněji. Vytvořil jsem si samostatný script pro tutoriál, pak jsem si vytvořil jednu stránku s ovládacími prvky a v té jsem jenom vyměňoval část s obrázkem a textem.

- Přidat do agendy Plán Směn možnost barevného šrafování a indikaci dostupnosti a poznámky u jednotlivých dnů.

V Plánu směn se vybarvují jednotlivé dny podle typu naplánované směny barvou, která mi chodí ze serveru. Bylo však nutné graficky rozlišit dny, na které má uživatel naplánováno více směn (například ranní a odpolední). To mělo zajistit šrafování ve kterém se střídaly barvy, svázané s konkrétními typy směn. Dále jsem měl zobrazovat v buňce kalendáře ikonku, pokud u naplánované směny byla nějaká poznámka a zelený nebo červený puntík v případě, že zaměstnanec oznamuje, kdy je nebo není dostupný a směny mu pak podle toho volí manažer.

- A dále řada drobnějších úprav GUI i chování aplikace a oprav chyb.

Tabulka 3: Odhad času stráveného na jednotlivých úkolech v aplikaci Moje Docházka

Úkol	Strávený čas [h]
Aktualizovat aplikaci pro novou verzi frameworku Onsen UI	40
Opravit zobrazování trasy služební cesty	20
Vytvořit aktivitu počítadlo, poznámka a cena	10
Stahovat sumy aktivit a projektů novou Rest API metodou	10
Vytvořit obrazovku s historií záznamů	10
Přepracovat tutoriál	5
Rozšířit Plán směn	5

### 3.4 Další úkoly

Dále jsem dělal:

- Tvorba překladů z textových řetězců umístěných přímo v kódu.

Tyto řetězce jsem nahradil funkcí, která mi podle systémového jazyka vrátila správný překlad a následně jsem je umístil do odpovídajících souborů s překlady.

- Oprava utility pracovně nazvané „Server Checker”.

Tato utilita v pravidelných intervalech komunikovala se serverem, a pokud zjistila zvýšení odezvy, abnormální využití procesoru nebo paměti RAM na serveru, tak okamžitě upozornila administrátora. Utilita ale náhodně falešně zobrazovala, že je server nedostupný. Proč se to stávalo se mi bohužel nepodařilo zjistit, hlavně kvůli náhodnosti problému, který nešel spolehlivě reprodukovat, ale opravil jsem pár chyb, kterých jsem si všiml. Jako například nefunkční ukládání barvy a infa do lokálního úložiště a hromadění dialogů v DOM, jelikož se dialogy vytvářely ale místo zrušení se pak jenom schovávaly.

Tabulka 4: Odhad času stráveného na dalších úkolech

Úkol	Strávený čas [h]
Vytvořit překlady pro texty v kódu	30
Opravit utilitu „Server Checker”	5

## 4 Závěr

### 4.1 Získané znalosti

Před praxí jsem měl pouze základní znalosti programovacího jazyka Java a HTML 4, které jsem si v průběhu dále rozšířil. V rámci praxe jsem si musel dostudovat principy vývoje aplikací pro systém Android, a dále se seznámit s vývojem webových aplikací v programovacím jazyce JavaScript v kombinaci s HTML 5 a CSS. Tyto znalosti jsem pak paralelně využíval i v předmětu Tvorba aplikací pro mobilní zařízení I. Musel jsem si nastudovat jak pracovat s frameworky AngularJS a Onsen UI mimo jiné. Vyzkoušel jsem si verzovací systémy Apache Subversion[14] i Git[15] a pochopil jsem, jak nezbytné jsou při práci v týmu. Objevil jsem také užitečnou online službu Trello[16] na organizaci a rozdělování úkolů v týmu. Vzhledem k faktu, že většina studijních materiálů, ze kterých jsem čerpal, je psaná v anglickém jazyce, bylo nezbytné rozumnět odbornému textu psanému v tomto jazyce.

Ze znalostí získaných ve škole jsem využil již zmíněný programovací jazyk Java z předmětu Programovací jazyky I (PJ I), pak konstrukce, datové struktury a polymorfni přiřazení z předmětů Algoritmy I a II (ALG I, II) a dále jsem měl možnost, podílet se na vývoji reálného 3 vrstvého informačního systému s několika různými prezentačními vrstvami, popisovaného v předmětu Vývoj informačních systémů (VIS). Také už jsem zmínil, že paralelně s praxí jsem studoval předmět Tvorba aplikací pro mobilní zařízení I (TAMZ I), který doplnil mé čerstvě nabyté znalosti vývoje webových aplikací a hned jsem je využil při vývoji aplikací na praxi.

Při samotném vypracování bakalářské práce jsem pak uplatnil znalosti z předmětu Softwarové inženýrství (SWI) při tvorbě diagramu komponent, viz. obrázek 2. Naopak jsem litoval, že jsem si nezapsal předmět Elektronické publikování (ELP), protože jsem doposud neměl žádné zkušenosti s typografickým systémem  $\text{\LaTeX}$  a před samotným započítím vypracování bakalářské práce jsem se s ním musel nejprve seznámit. Nicméně jakožto dosavadní uživatel textového editoru Microsoft Word musím říct, že se mi líbí více programátorský přístup systému  $\text{\TeX}$  a pro odborné publikace ho budu používat mnohem radši.

### 4.2 Zhodnocení praxe

Množství nových technologií jsem se sice zpočátku děsil, ale nakonec jsem rád, že jsem se s nimi seznámil a věřím, že se mi nabyté vědomosti a zkušenosti určitě budou hodit v dalším studiu a následně i v zaměstnání. Celkově svou bakalářskou praxi hodnotím kladně, protože jsem měl možnost nahlédnout do vývoje běžného komerčního produktu, a taky jsem se díky tomu mohl naučit spoustu věcí, ke kterým bych se jinak ve škole nedostal. Praxe mi také pomohla udělat si lepší představu o tom, jakým směrem bych se chtěl dále ubírat.

Mám-li zhodnotit další vývoj předmětu práce, tak jde bezesporu o perspektivní produkt který v mnoha ohledech překonává konkurenci a i díky vstřícné politice firmy získává samé

kladné ohlasy. Proto si myslím že se nadále bude dostávat do povědomí lidí, a získávat nové klienty, kvůli kterým se vyplatí pokračovat ve vývoji.

Pro informaci uvádím, že tato práce souvisí s prací mého kolegy, Petra Burdíka, který absolvoval odbornou praxi v téže firmě ve stejném roce.



## Literatura

- [1] GIRITON Systems s.r.o. Online Docházka GIRITON [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://giriton.com/>
- [2] AngularJS. AngularJS - Superheroic JavaScript MVW Framework [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://angularjs.org/>
- [3] Onsen UI. Onsen UI: Beautiful HTML5 Hybrid Mobile App Framework [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://onsen.io/>
- [4] Apache Cordova. Apache Cordova [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://cordova.apache.org/>
- [5] Android Developers. FloatingActionButton | Android Developers [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://developer.android.com/reference/android/support/design/widget/FloatingActionButton.html>
- [6] Font Awesome. Font Awesome, the iconic font and CSS toolkit [online]. 2016 [cit. 2016-04-19]. Dostupné z: <http://fontawesome.io/>
- [7] Android Developers. RippleDrawable | Android Developers [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://developer.android.com/reference/android/graphics/drawable/RippleDrawable.html>
- [8] GitHub. GitHub - balysv/material-ripple: Android L Ripple effect wrapper for Views [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://github.com/balysv/material-ripple>
- [9] Android Central. Everything you need to know about rooting your Android | Android Central [online]. 2016 [cit. 2016-04-19]. Dostupné z: <http://www.androidcentral.com/root>
- [10] Google Play. Google Play [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://play.google.com/store?hl=en>
- [11] World Wide Web Consortium (W3C). Carousel Concepts • Carousels • WAI Web Accessibility Tutorials [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://www.w3.org/WAI/tutorials/carousels/>
- [12] Electron. Electron - Build cross platform desktop apps with JavaScript, HTML, and CSS [online]. 2016 [cit. 2016-04-19]. Dostupné z: <http://electron.atom.io/>
- [13] Slick. Slick - the last carousel you'll ever need [online]. 2016 [cit. 2016-04-19]. Dostupné z: <http://kenwheeler.github.io/slick/>
- [14] Apache Subversion. Apache Subversion [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://subversion.apache.org/>

[15] Git. Git [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://git-scm.com/>

[16] Trello. Trello [online]. 2016 [cit. 2016-04-19]. Dostupné z: <https://trello.com/>